Aspectos a considerar sobre los lenguajes de implementación

Es uno de los objetivos de la materia que aprendan a modelar problemas mediante estructuras de datos complejas, a resolverlos con los algoritmos apropiados y entender y expresar las complejidades asociadas en aspectos teóricos y prácticos.

El lenguaje a utilizar debe favorecer lo anterior mediante

- una interfaz que facilite el modelado.
  Ej. reflexiva sobre las estructuras del lenguaje, entendiendo las unidades conceptuales de modelado, como clase, metodo, mensaje, instancia etc.
- una interfaz con herramientas que permitan el uso de metodologías ágiles.
  Ej. framework de Unit Tests
  Ej. permitiendo el desarrollo incremental.
- una interfaz que facilite la exploración del modelo
  Ej. mediante debugging, permitiendo el seguimiento del algoritmo; cambios en caliente; múltiples hilos de ejecución a partir de un estado; start, stop y restart del thread en ejecución.
- una implementación uniforme
  Ej. Tipos Primitivos, Objetos, Clases
- una Metáfora de programación apropiada
  Ej. Orientada al hardware vs. Orientado al humano
- una implementación que permita verificar las especificaciones y de código abierto
- una separación entre el modelo y la máquina apropiado.
  Ej: Administración de memoria. Tipos de datos.
- Es deseable que el lenguaje sea dinámicamente y fuertemente tipado.
- Sin distinción entre los tipos propios y nativos. La homogeneidad favorece el uso de tipos de alto nivel desarrollados por usuario.

	C++	Java	Smalltalk
Calidad de la IDE	-	+/-	+
Interfaz reflexiva	-	+/-	+
Desarrollo incremental	-	+	+
Debugging	-	+/-	+
Cambios en Caliente	-	+/-	+
múltiples threads a partir de un punto	-	-	+
start/stop/restart	-	+/-	+
implementación uniforme	-	+/-	+
metáfora de programación	-	+/-	+
legibilidad del código de terceros	-	+/-	+
UnitTests	+/-	+/-	+
Administración de memoria	-	+	+
Tipado	Estático	Estático*	Dinámico
Tipado	Débil	Fuerte	Fuerte
Homogeneidad de la IDE y lenguaje	-	-	+

* Si bien los tipos de los objetos se definen en tiempo de compilación, Java posee extensiones y mecanismos de reflexión que le permite comportarse dinámicamente, pero con un costo importante en términos de legibilidad y claridad

Estos tres lenguajes podrían considerarse como exponentes paradigmáticos de los lenguajes, C++ orientado al hardware y de bajo nivel. Java como algo intermedio. Y Smalltalk orientado a la persona y de alto nivel.

El primero se propone por la familiaridad que ya han logrado a lo largo de la carrera a pesar de sus desventajas como lenguaje. El segundo por su similaridad con C++ y porque carece de la mayoría de sus inconvenientes pedagógicos. El tercero porque cumple con todas las condiciones buscadas.

La cátedra propone el uso de Java como lenguaje de implementación por ser algo intermedio entre lo familiar y la novedad pero con buena parte de las ventajas.
No obstante, los tres lenguajes van a ser soportados desde la cátedra y se dará asistencia.

Java 6 o 7

(Pueden usar lo que esté en los laboratorios)
new to java http://www.oracle.com/technetwork/topics/newtojava/overview/index.html
java 7 Standar Edition http://www.oracle.com/technetwork/java/javase/overview/index.html
JDK Java Developement Kit http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html
JRE Java Runtime Enviroment
http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

Eclipse Mars(requiere mínimo java 7 JRE instalado)
http://www.eclipse.org/users/
https://eclipse.org/downloads/packages/eclipse-ide-java-developers/marsr (170Mb)

Descargar el archivo zip y expandir en el directorio donde se desea que quede instalado
revisar eclipse.ini
(-Dosgi.requiredJavaVersion, y reducir de ser necesario -Xms56m y -Xmx524m, o valores intermedios)
Ejecutar eclipse.exe
La primera vez preguntará dónde ubicar el workspace. Es la ubicación donde colocará la información sobre las configuraciones personales del usuario y sus proyectos de trabajo

Tutoriales y documentación
sobre la plataforma http://help.eclipse.org/luna/index.jsp?nav=%2F0

Instalación de tp Java de ejemplo

- (cerrar la ventana de bienvenida) pueden volver a abrirla en Help->Welcome
- en la vista Package Explorer, botón derecho -> import
- General -> Existing Projects into Workspace … Next
- select root directory el directorio donde se encuentre el archivo.project descomprimido…. Finish
- Sucesivos paquetes pueden agregarse desde la interfaz o desde el file system

Testing
en el menu de contexto de las clases SuiteXxxx o TestXxxx puede leejirse Run As -> JUnit Test.

Pharo

http://pharo.org/download/ (20Mb)

Descargar archivo zip y expandir en el directorio donde se desea que quede instalado
Ejecutar pharo.exe

Tutorial sobre uso de interfaz
Al descargar y abrir pharo por primera vez hay un tutorial muy breve 'PharoTutorial go' con las instrucciones para correrlo.

Instalación de tp Pharo de ejemplo

- Copiar el contenido de TpEjemplo.pharo.zip en el directorio de instalacion de Pharo
- Drag & drop de EFO-ALGO3-tp.st sobre la ventana de Pharo. Aceptar 'FileIn entire file'
- Abrir un Class Browser: Tocan el fondo con el mouse y en el menú de contexto elegir System Browser, buscan la categoría (primer lista) EFO-Algo3 y observar el código.

Testing
Los botones en gris sobre las Clases o sobre los métodos permiten correr los test. Tocandolos con el mouse pasarán a verde, amarillo o rojo según el resultado de la corrida del test.

Contenido de los archivos de test TP1

Ej1:

6	
6 8 12 15	3
35	
8 14 20 40 45 54 60 67 74 89 99	6
100	
35 87 141 163 183 252 288 314 356 387	4
90	
6 8 16 19 28 32 37 45 52 60 69 78 82	14
4	
5 13 19 26 35	0 // no hay conexión posible
5	
5 13 19 26 35	2 // comunica la capital con la primer estación
5	
7 16 19 27 33	2
8	
2 5 8 14 18	4 // desde la capital al kmt 8
8	
3 6 9 15 19	3

ej 2:

2 3 4 1 2	2 2 3 2 2
2 7 2 8 4 9 1 6 5	2 4 2 4 4 5 4 5 5
1 87 4	1 44 4
4 0 32 6 8 10	4 2 4 5 6 7

ej 3 :

a bcde;b acde;c abde;d abc;e abc	2 abdce
a bcd;b ae;c ad;d ac;e b	2 abecd
a fb;b gc;d gc;f agh;e hd	3 abgcdehf
x yz	1 xyz

Contenido de los archivos de test TP2

Ej1:

Entrada/Salida de ejemplo

10 // pisos en el pabellón	1
0 10 // un sólo portal desde la planta baja al 10mo piso	
9	2
0 5; 1 6; 2 7; 5 9	

Formato de entrada

```
n                // entero
pd ph[;p p]      // pd (piso desde), ph (piso hasta) enteros, pares de p separados por ;
```

Salida:

```
n                // cantidad de portales
```

Ej 2:

Entrada/Salida de ejemplo

10 10	21
0 10 10 1	
4 4	18
0 1 1 2; 1 2 3 1; 2 3 4 0; 3 4 2 1	

Formato de entrada

```
n m                              // pisos y metros de cada piso
pd md ph mh[;pd md ph mh]  // pd (piso desde), md (metros desde),
                                 // ph (piso hasta), mh (metros hasta) enteros, separados por ;
```

Salida:

```
n                          // cantidad de segundos
```

Ej 3 :

Entrada/Salida de ejemplo:

0 1 3; 1 2 3; 2 0 3	3
0 1 8; 0 4 70; 0 3 63; 1 2 53; 1 4 54; 2 3 10; 2 4 12; 3 4 22	52

Formato de entrada

```
id ih l[;id ih l]          // (intersección desde) (intersección hasta) (longitud)
```

Salida:

```
n                // Suma mínima de las longitudes de los pasillos que se deben clausurar.
```